

WHAT IS CLAIMED IS:

1. In a computer system having a memory for storage of files and a main filesystem for accessing said files, the improvement of adding a custom filesystem, said custom filesystem comprising:  
a custom hierarchical structure of files and folders; and  
a set of links between said custom files and the locations of corresponding real files in said main file system;  
said custom file system having priority over said main file system, in that a process manager will attempt to address requests on said custom filesystem prior to placing said requests on said main filesystem.
2. The computer system of claim 1, wherein said custom hierarchical structure is generated by determining which of said files in said main filesystem should be viewed.
3. A method of preparing a custom filesystem for a computer system, comprising the steps of:  
generating and storing a file of attributes for said computer system;  
selecting required software applications from available software applications in accordance with said attributes; and  
linking said selected software applications to said custom filesystem.
4. The method of claim 3, wherein said system's attributes include static characteristics.
5. The method of claim 4, wherein said static characteristics include a microprocessor model and an operating system platform.
6. The method of claim 3, wherein said system's attributes include dynamic characteristics.
7. The method of claim 4, wherein said dynamic characteristics include preferences regarding accessible software applications and versions of said software applications.

09824255.040304

8. The method of claim 7 wherein said step of responding comprises the steps of:
- for each available software application, determining whether said system requires said software application, and if so:
- querying said system regarding a desired version of said software application; and
- adding said desired version of said software application to said custom filesystem.
9. The method as claimed in claim 7, wherein said dynamic characteristics include:
- the location of the package repositories;
- the packages to include;
- the root of the custom file system; and
- the spill directory root.
10. The method as claimed in claim 7, wherein said attributes include:
- a Root directory to take over [cfsroot];
- a Directory to copy changed (spilled) files to [spillroot];
- a Directory locations (base root for the packages); and
- a Packages to read and customize.
11. The method as claimed in claim 10, wherein said linking is symbolic linking.
12. The method as claimed in claim 10, comprising the step of:
- storing an image of the current custom file system which may be re-generated, allowing roll-back.
13. The method as claimed in claim 4, comprising the step of:
- responding to a request to roll-back by replacing said custom file system with said image of said custom file system.
14. A method of file system management comprising:
- determining which files a system will see based on system attributes;
- identifying filenames representing said files said system will see; and

generating links between said filenames and real software locations of said files said system will see;  
said identified filenames and links comprising a custom filesystem.

15. A method of preparing a custom filesystem for a computer system, comprising the steps of:  
generating and storing a file of attributes for said computer system in a system configuration file;  
processing said system configuration file;  
reading all directories in a main file system to generate a search path;  
traversing said search path and for each available software application, determining whether said system requires said software application, and if so:  
querying said system regarding a desired version of said software application; and  
adding said desired version of said software application to said custom filesystem.
16. A custom filesystem comprising:  
means for generating and storing a file of attributes for said computer system;  
means for selecting required software applications from available software applications in accordance with said attributes; and  
means for linking said selected software applications to said custom filesystem.
17. A method of operation for a custom filesystem comprising the steps of:  
loading a system configuration file;  
responding to receipt of a request to perform an operation by:  
accessing said system configuration file;  
re-directing said requested operation to a corresponding real file location; and  
performing said requested operation with respect to said real file.
18. The method of claim 17, further comprising the step of loading packages onto custom pathname tree, subsequent to said step of loading a system configuration file.

19. The method of claim 18, wherein said system configuration file comprises static and dynamic system characteristics of said system.
20. The method of claim 19, further comprising the step of responding to a request from the system by:  
responding to a targeted file on the package file system being invalid by sending an error message to said end system.
21. The method of claim 20, further comprising the step of responding to a request from the system by:  
responding to the state of said targeted file not being known by updating the state of said targeted file.
22. The method as claimed in claim 17, further comprising the step of:  
responding to a command which requires modification of a file by copying said file to a spill directory tree.
23. The method of claim 21, further comprising the step of responding to a request from the system by:  
responding to said file having been spilled by requiring request spilled location.
24. A method of operation for a custom filesystem comprising:  
loading system configuration file;  
loading packages onto custom pathname tree;  
responding to a request from the system by:  
pulling out filtering characteristics;  
responding to a targeted file on the package file system being invalid by sending an error message to said end system;  
responding to the state of said targeted file not being known by updating the state of said targeted file;  
responding to said file having been spilled by requiring request spilled location;  
responding to said request being a read request by:

re-directing said read request to a corresponding "real" file location;  
and  
performing said read operation;  
responding to said request being a write request by:  
marking said target file in said main root directory;  
creating said spillroot direction; and  
copying target file to said spill root location; and  
responding to said request being a stat request by accessing meta-data from  
package file location as required.

25. A carrier signal incorporating software code executable to perform the method steps of:  
generating and storing a file of attributes for said computer system;  
selecting required software applications from available software applications in  
accordance with said attributes; and  
linking said selected software applications to said custom filesystem.
26. A computer readable memory medium for storing software code executable  
to perform the method steps of:  
generating and storing a file of attributes for said computer system;  
selecting required software applications from available software applications in  
accordance with said attributes; and  
linking said selected software applications to said custom filesystem.
27. A method of file management comprising the steps of:  
displaying a customized file system in which filenames are linked to real software file  
locations in accordance with a system's attributes; and  
responding to an input of an instruction with respect to said customized file system  
by:  
re-directing said instruction to said real software file location.
28. A computer system comprising:  
a central processing unit (CPU);  
an input/output unit; and  
a memory;

